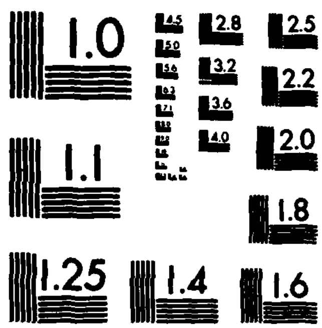
AD-A148 925 COORDINATION IN A DISTRIBUTED PROBLEM SOLVING NETWORK (U) MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER AND INFORMATION SCIENCE D D CORKILL ET AL. 1984 1/1

UNCLASSIFIED NSF-MCS80-06327 F/G 5/1 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC
SELECTE
DEC 3 1 1984
S D
E
1984
(2)
WA4
AD-A148 925

COORDINATION IN A DISTRIBUTED PROBLEM SOLVING NETWORK

Daniel D. Corkill and Victor R. Lesser
Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts, 01003

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

ABSTRACT

Distributed problem solving networks provide an interesting application area for high-level network coordination through the use of organizational structuring. We describe a decentralized approach to the coordination of these networks that relies on each node making sophisticated local activity decisions. Each node is guided by a high-level strategic plan for cooperation among nodes in the network and must balance its own perceptions of appropriate problem solving activity with activities deemed important by other nodes. The high-level strategic plan, which is a form of meta-level control, is represented as a network organizational structure specifying in a general way the information and control relationships among the nodes. In addition to its application to Distributed Artificial Intelligence, this research has implications for organizing and controlling complex knowledge-based systems that involve semi-autonomous problem solving agents.

INTRODUCTION

Cooperative distributed problem solving networks are distributed networks of semi-autonomous nodes (processing elements) that are capable of sophisticated problem solving and cooperatively interact with other nodes to solve a *single* problem. Each node can itself be a sophisticated *problem solving system*, that can modify its behavior as circumstances change and plan its own communication and cooperation strategies with other nodes.

A key problem in cooperative distributed problem solving networks is obtaining sufficient global coherence for effective cooperation among the nodes [SMIT81]. If this coherence is not achieved, then the performance (speed and accuracy) of the network can be significantly diminished as a result of:

1. This research was sponsored, in part, by the National Science Foundation under Grant MCS-8006327 and by the Defense Advanced Research Projects Agency (DOD), monitored by the Office of Naval Research under Contract NR049-041.

DTIC FILE COPY

84 10 10 168

- lost processing as nodes wait for something to do;
- wasted processing as nodes work at cross-purposes with one another;
- redundantly applied processing as nodes duplicate efforts;
- misallocation of activities so that important portions of the problem are either inaccurately solved or not solved in timely fashion.

Network coordination is difficult because limited internode communication restricts each node's view of network problem solving activity. In addition, network reliability issues (which require that the network's performance degrades gracefully if a portion of the network fails) preclude the use of a global "controller" node. Instead, each node must be able to direct its own activities in concert with other nodes, based on incomplete, inaccurate, and inconsistent information. This requires a node to make sophisticated local decisions that balance its own perceptions of appropriate problem solving activity with activities deemed important by other nodes.

As will be discussed later, such node sophistication is an important requirement for effective cooperation among large numbers of nodes operating in dynamic distributed problem solving environments. First, however, we describe the problem solving approach we have developed for cooperative distributed problem solving networks.

THE FUNCTIONALLY ACCURATE, COOPERATIVE APPROACH

We have been designing cooperative problem solving networks for applications in which there is a natural spatial distribution of information and processing requirements, but insufficient information for each processing node to make completely accurate control and processing decisions without extensive internode communication (used to acquire missing information and to determine appropriate node activity). An example of this type of application is a distributed sensor network [LACO78, SMIT78, LESS80b]. In a distributed sensor network the data received by a node from its sensors is highly error prone and approximate. Therefore, a node cannot generate an accurate interpretation of its sensory data without cooperation with other nodes to obtain a view of their sensory data.

Our design approach for implementing these applications as distributed networks is use cooperation among nodes so that the network as a whole can function effectively even though the nodes have inconsistent and incomplete views of the information used in their computations. We call this type of distributed problem solving network *functionally accurate, cooperative* (FA/C) [LESS81, LESS82]. In the FA/C approach, the distributed network is structured so that each node can perform useful processing using incomplete input data, while simultaneously exchanging partial, tentative intermediate results of its processing with other nodes to construct cooperatively a complete solution. The hope is that the amount of communication required to exchange these results is much less than the communication of raw data and processing results which

would be required using a conventional distributed processing approach. In addition, the synchronization required among nodes can also be reduced, resulting in increased node parallelism and network robustness. As a result of our previous experimental work with a distributed version of the Hearsay-II speech understanding system [LESS80c], our current work on a distributed vehicle monitoring network [LESS82], and on a distributed network traffic light control network [BROO83], we have shown that these hopes can, in fact, be realized.

THE IMPORTANCE OF NETWORK COORDINATION

A key problem in the successful application of the FA/C approach (and which we feel is a major issue in the design of cooperative distributed problem solving networks) is obtaining a sufficient level of cooperation and coherence among the activities of the semi-autonomous nodes in the network. Coordination problems arose in the distributed version of the Hearsay-II speech understanding system (a rudimentary FA/C system) in which nodes interacted through the exchange of a small number of high-level hypotheses and each node determined locally what work it should perform and information to transmit. The *data-directed* and *self-directed* control regime used in experiments with a three node speech[1] understanding network lead to non-coherent behavior [LESS80a]. Situations occurred when a node had obtained a good solution in its portion of the overall utterance and, having no way to redirect its attention to new problems, the node simply produced alternative but worse solutions.

Another problem occurred when a node had noisy data and could not possibly find an accurate solution without help from other nodes. In this situation, the node with noisy data often quickly generated an inaccurate solution which, when transmitted to nodes working on better data, resulted in the distraction of these nodes. We believe that development of appropriate network coordination policies (the lack of which resulted in diminished network performance for even a small network) will be crucial to the effective construction of large distributed problem solving networks containing tens and hundreds of processing nodes.

This type of network coordination is significantly different from distributed task allocation techniques developed for conventional distributed processing systems. In FA/C problem solving networks, the network is working on a single problem that is potentially solvable in many different ways. Many of the potential tasks are either unnecessary (because they work with overlapping or independent views of data that has already been processed in another task) or inappropriate (because the solution is being developed in a different way). Since only a subset of the possible tasks need to be executed, network coordination is more akin to focus of attention techniques used to control search in Artificial Intelligence systems [HAYE77] than to the task scheduling problem addressed by conventional distributed task allocation techniques.

In order for a network coordination policy to be successful, it must achieve the following conditions:

coverage – any given portion of the overall problem must be included in the activities of at least one node;

connectivity – nodes must interact in a manner which permits the covering activities to be developed and integrated into an overall solution;

capability – coverage and connectivity must be achievable within the communication and computation resource limitations of the network.

It is important that the network coordination policies do not consume more processing and communication resources than the benefits derived from the increased problem solving coherence. We believe that in networks composed of even a small number of nodes, a complete analysis to determine the detailed activites at each node is impractical. The computation and communication costs of determining the activities far outweigh the improvement in problem solving performance. Instead, coordination in distributed problem solving networks must sacrifice some potential improvement for a less complex coordination problem.

What is desired is a balance between problem solving and coordination so that the combined cost of both activities are acceptable. The emphasis is shifted from optimizing the activities in the network to achieving an acceptable performance level of the network as a whole. These policies must also be appropriately flexible that they provide sufficient network robustness and reliability to respond to a changing task and hardware environment.

This approach to network coordination which emphasizes finding an acceptable range of behavior rather than optimal is similar to most human problem solving (both individual and organizational) where the concern is with achieving a satisfactory performance level rather than an optimal one [MARC58]. Termed *satisficing*, this level of performance can be significantly less complex than optimizing. Determining if the activities in the network are optimal requires:

- o a set of criteria permitting all alternative sequences of network activities to be compared;
- o using these criteria to decide whether the particular sequence of network activities is preferred to all the alternatives;

while determining if the activities are satisfactory requires:

- o a set of criteria describing minimally satisfactory performance levels;
- o using these criteria to decide whether the particular sequence of network activities is minimally satisfactory.

March and Simon compare optimizing to "searching a haystack to find the *sharpest*

needle" and satisficing to "searching the haystack to find a needle sharp enough to sew with".

In order for network coordination to satisfy these requirements of reasonable cost and of flexibility, it must be able to tolerate the lack of up-to-date, incomplete, or incorrect coordination information due to delays in the receipt of information, the high cost of acquisition and processing of the information, or errors in communication and processing hardware. Network coordination of this form is very similar in concept to the type of local control necessary in FA/C networks where it is assumed there is always some level of coordination uncertainty and the goal is not an optimal answer but one within an acceptable range.

NETWORK COORDINATION VIA ORGANIZATIONAL STRUCTURING

The interplay between local node control and network-wide control is a crucial aspect of the design of decentralized network coordination policies. It is unrealistic to expect that network coordination policies can be developed which are sufficiently flexible, efficient, and require limited communication, while simultaneously making all the control decisions for each node in the network. A node needs a sophisticated form of local control that permits it to plan sequences of activities and to adapt its plan based on its problem solving role in the network, on the status and role of other nodes in the network, and on self-awareness of its activities. Using such sophisticated local node control, a wide range of network coordination policies can be developed which balance externally-directed control (needed for network coherence) with self-directed control (needed for quick adaptation to changing conditions and limited communication requirements).

One of the ways that sophisticated and self-aware local node control can be exploited is to split the network coordination problem into two concurrent activities [CORK83]:

1. construction and maintenance of a network-wide organizational structure;
2. continuous local elaboration of this structure into precise activites using the local control capabilities of each node.

The organizational structure specifies the information, communication, and authority relationships among the nodes in only a very general way. Included in the organizational structure are control decisions that are not quickly outdated and that pertain to a large number of nodes. The organizational structure represents general "ballpark" control decisions which are tailored by the local node control.

In a sense, an organizational structure is a high-level "strategic" plan describing and delimiting the gross responsiblities of each node in the network. A significant portion of the control activity of each node is elaboration of these responsibilities into precise activities to be performed by the node. For example, in a simple hierarchical organization, each low-level "worker" node must still decide what particular activities

are required to satisfy its responsibilities and determine what particular information should be passed up to the higher-level "integrating" node and laterally to other interested worker nodes (as specified by the integrating node).

The existence of an organizational structure provides a control framework which reduces the amount of control uncertainty present in a local node (due to incomplete or errorful local control information) and increases the likelihood that the nodes will be coherent in their behavior by providing a general strategy for network problem solving that is available to all nodes. The organizational structuring approach to limiting control uncertainty still preserves a certain level of control flexibility for a node to adapt its local control to changing task and environmental conditions and to inappropriate external direction.

The use of organizational structuring as means of network coordination naturally leads to the idea of dynamic modification of the organizational structure. An inflexible organizational structure can lead to a loss of network effectiveness if the internal or external environment of the distributed problem solving network changes. For example, in a one-level hierarchical organizational structure, worker responsibilities may need to be reallocated if some worker nodes are overloaded and other worker nodes remain relatively idle. If the integrating node becomes overloaded, additional non-local decision-making authority may need to be passed down to the worker nodes. If the integrating task becomes excessively difficult, the entire hierarchical structure may need to be augmented with an intermediate level of integrating nodes. It may even be appropriate to replace the hierarchical structure with a completely different organizational form.

Because an effective organizational structure is dependent upon the dynamics of the problem solving situation, the distributed problem solving network must initially develop an organizational structure and as problem solving progresses:

- o monitor for decreased effectiveness caused by an inappropriate organizational structure;
- o determine plausible alternative structures;
- o evaluate the cost and benefits of continuing with its current structure versus reorganizing itself into one of the alternative structures;

This development and maintenance of an organizational structure by the network itself is ***organizational self-design***.

There are two basic approaches to organizational self-design. One approach is to predetermine a "cookbook" of problem solving situation-organization pairs. The network monitors for a change in its problem solving situation and, if a change is detected, uses the associated predetermined organizational structure as its new organizational form. A second approach is to provide the network with knowledge about situations and organizational forms and have the network develop plausible

alternative structures as the situation warrants. An orthogonal issue is how the network performs organizational self-design. The design task could be performed at a single "designer" node. Alternatively, the organizational design task could itself be distributed among the nodes, proceeding concurrently with the overall problem solving task.

Two additional components are also relevant to the organizational structuring approach to network coordination:

- A *distributed task allocation* component for deciding what information and requests should be transmitted among the nodes. Given the high-level strategic plan for the allocation of activities and control responsibilities among nodes (the organizational structure) there is still a need to make more localized, tactical decisions that balance the activities among the nodes based on the dynamics of the current problem solving situation [PAVL83].
- A *knowledge-based fault-diagnosis* component for detecting and locating inappropriate system behavior. We are looking to not only isolate problems caused by hardware errors, but also inappropriate settings of the problem solving parameters that specify strategic and tactical network coordination [HUDL83].

THE DISTRIBUTED VEHICLE MONITORING TESTBED: A TOOL FOR INVESTIGATING NETWORK COORDINATION

A large part of our work has been the construction of an appropriate experimental environment that will permit the exploration of alternative approaches for high-level network coordination. The results of these efforts has been the development of the Distributed Vehicle Monitoring Testbed: a flexible and fully-instrumented research tool for the empirical evaluation of alternative distributed problem solving network designs [LESS82]. The testbed, which is now fully operational, can be used to explore not only the use of organizational design for network coordination but also other approaches such as the use of negotiation among nodes (a key element of the contract network formalism [SMIT78]), distributed load balancing, and local planning of when and how to communicate with other nodes.

The testbed simulates a network of problem solving nodes attempting to identify, locate, and track patterns of vehicles moving in a two-dimensional space based on signals detected by accoustic sensors. Each node is an architecturally-complete Hearsay-II-like system, extended to include more sophisticated local control through the addition of a planning module and capabilities for communication of goals [CORK81, CORK82]. The planner can adapt local node activity in response to a node's current organizational roles, externally-directed requests by other nodes (communicated goals), and the potential processing activities of the node (based on the data it is receiving from its local sensors and from other nodes and the results it has so far produced).

The planner is highly parameterized. One important parameter varies the priority given to achieving the node's organizational roles versus satisfying goals received from another node or satisfying the goals generated internally as a result of the node's own processing. The organizational roles of a node are specified in a non-procedural manner through a data structure called the *organizational blackboard* that can be adjusted dynamically. The following factors are used to define the organizational roles of a node:

- the organizational importance of having the node generate hypotheses at particular blackboard levels, times, spatial areas, and event classes;[1]
- the organizational importance of having the node send or accept hypotheses and goals to or from particular nodes at particular blackboard levels, times, areas, and event-classes.

By varying the parameters of the planner and the organizational roles of nodes, a wide variety of static network architectures and coordination policies can be evaluated [CORK83]. Likewise, the non-procedural specification of the node's organization role permits an organizational design component to be easily added to the testbed in order to explore the concept of organizational self-design.

SUMMARY AND FUTURE RESEARCH

Effective network coordination is an important problem in the use of cooperative distributed problem solving networks. We have described an approach to network coordination through the use of organizational structuring. The organizational structure provides each node with a high-level view of problem solving in the network. The sophisticated local control component of each node is responsible for elaborating these relationships into precise activities to be performed by the node, based on the node's problem solving role in the network, on the status and organizational roles of other nodes in the network, and on self-awareness of the node's activities. The balance between local node control and organizational control is a crucial aspect of this approach.

We have developed a node architecture capable of the sophisticated local decisionmaking necessary for balancing the node's perceptions of appropriate problem solving activity with activities deemed important by other nodes. We have implemented this node architecture in the Distributed Vehicle Monitoring Testbed: a flexible research tool for the empirical evaluation of distributed network designs and coordination policies. Our ongoing research builds on this node architecture and the testbed to explore (through actual implementation and empirical studies) how different

1. An event class specifies a particular vehicle type or characteristic of a vehicle, depending on the blackboard level of the hypothesis.

organizational policies perform in various problem solving situations. One goal of this research is the development of a distributed organizational self-design component in the testbed.

It is interesting to note that the themes of this research, which advocate the interplay between organizational control and sophisticated local node control, are close in emphasis to recent trends emphasizing meta-level control and sophisticated planning in knowledge-based Artificial Intelligence systems [HAYE79, DAVI80, STEF80, ERMA81]. As Nilsson and Erman have noted, the field of distributed Artificial Intelligence serves to illuminate basic Artificial Intelligence issues [NILS80, ERMA82]. In this case, the need to control the uncertainty inherent with semi-autonomous problem solving agents possessing only a local and possibly errorful view of network problem solving activity is very similiar to the control problems that are being faced in the development of the new generation of knowledge-based problem solving systems which have significantly larger and more diverse knowledge bases.

REFERENCES

BROO83 Richard Samuel Brooks.
Experiments in Distributed Problem Solving with Iterative Refinement.
PhD Thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, February 1983.
Available as Technical Report 82-25, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, October 1982.

CORK81 Daniel D. Corkill and Victor R. Lesser.
A goal-directed Hearsay-II architecture: Unifying data and goal directed control.
Technical Report 81-15, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, June 1981.

CORK82 Daniel D. Corkill, Victor R. Lesser, and Eva Hudlicka.
Unifying data-directed and goal-directed control: An example and experiments.
In *Proceedings of the Second National Conference on Artificial Intelligence*, pages 143-147, August 1982.

CORK83 Daniel David Corkill.
A Framework for Organizational Self-Design in Distributed Problem Solving Networks.
PhD Thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, February 1983.
Available as Technical Report 82-33, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, December 1982.

DAVI80 Randall Davis.
Meta-rules: Reasoning about control.
Artificial Intelligence 15(3):179-222, December 1980.

ERMA81 Lee D. Erman, Philip E. London, and Stephen F. Fickas.
The design and an example use of Hearsay-III.
Proceedings of the Sixth International Joint Conference on Artificial Intelligence, pages 409-415, August 1981.

ERMA82 Lee D. Erman.
Comments at 1982 Workshop on Distributed Artificial Intelligence.
USC Conference Center, Idyllwild, California, June 1982.

HAYE77 Frederick Hayes-Roth and Victor R. Lesser.
Focus of attention in the Hearsay-II speech understanding system.
In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pages 27-35, August 1977.

HAYE79 Barbara Hayes-Roth and Frederick Hayes-Roth.
A cognitive model of planning.
Cognitive Science 3(4):275-310, October-December 1979.

HUDL83 Eva Hudlicka and Victor R. Lesser.
A knowledge-based approach to fault diagnosis.
Technical Report, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, in preparation.

LACO78 R. Lacoss and R. Walton.
Strawman design of a DSN to detect and track low flying aircraft.
Proceedings of the Distributed Sensor Nets Workshop, pages 41-52, December 1978.
(Copies may be available from the Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 15213.)

LESS80a Victor R. Lesser, Jasmina Pavlin, and Scott Reed.
Quantifying and simulating the behavior of knowleldge-based interpretation systems.
Proceedings of the First National Conference on Artificial Intelligence, pages 111-115, August 1980.

LESS80b Victor R. Lesser.
Cooperative distributed problem solving and organizational self-design.
"Reports on the MIT Distributed AI Workshop", *SIGART Newsletter*, page 46, October 1980.
Also in the same issue: "Models of problem-solving," page 51.

LESS80c Victor R. Lesser and Lee D. Erman.
Distributed interpretation: A model and experiment.
IEEE Transactions on Computers, C-29(12):1144-1163, December 1980.

LESS81 Victor R. Lesser and Daniel D. Corkill.
Functionally-accurate, cooperative distributed systems.
IEEE Transactions on Systems, Man, and Cybernetics, SMC-11(1):81-96, January 1981.

LESS82 Victor Lesser, Daniel Corkill, Jasmina Pavlin, Larry Lefkowitz, Eva Hudlicka, Richard Brooks, and Scott Reed.
A high-level simulation testbed for cooperative distributed problem solving.
Proceedings of the Third International Conference on Distributed Computer Systems, pages 341-349, October 1982.

MARC58 James G. March and Herbert A. Simon.
Organizations.
Wiley, 1958.

NILS80 Nils J. Nilsson.
Two heads are better than one.
SIGART Newsletter (73):43, October 1980.

PAVL83 Jasmina Pavlin and Victor R. Lesser.
Task allocation in distributed problem solving systems.
Technical Report, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, in preparation.

END

FILMED

1-85

DTIC